

1. Introduction

Optimization in underground mine design has received less attention than that in open pit mines. This is mostly due to the diversity of underground mining methods and complexity of underground mining parameters, which do not allow the development of general optimization tools, especially, software tools or computer programs. However, few algorithms are available for optimization of ultimate stope boundaries, which are either tailored for a specific mining method (i.e. rigorous algorithms) or lack rigorous mathematical proof and fail to guarantee the true optimum (i.e. heuristic algorithms). Fewer software tools are available based on these algorithms.

The heuristic approaches include the Floating Stope technique of Datamine and the Maximum Value Neighborhood (MVN) algorithm. Both algorithms provide 3D analysis of the solution and can be applied to any underground mining methods. Nevertheless, they lack mathematical proof and fail to guarantee the true optimum solutions. All these algorithms apply on an economic block model of the ore body. Based on the Floating Stope technique a software tool has been developed by Datamine Software Company [1], which is the most popular and professional software applied to the problem. A non-commercial program called Stope Limits Optimizer (SLO) has been developed based on the MVN concept using Fortran 90 code. An application of the SLO for stope limits optimization on a hypothetical 3D model has been reported [2].

Rigorous algorithms include the Dynamic Programming solution, the Branch and Bound technique and the Probable Stope algorithm, which are developed for few underground mining methods and supported by mathematical reasoning. The dynamic programming technique provides a 2D analysis and has been adjusted only for block caving method. A Fortran IV code was prepared for this algorithm and has been run over some hypothetical models [3]. The branch and bound technique provides a 1D solution to the optimization of stope geometry and determines the start and the end of a stope in one direction. Several commercial, mathematical programming computer software packages provide Branch and Bound technique facilities. MPS and LINDO are two examples of the most popular packages [4]. An application of this technique has been reported by

MPS software for optimization of stope layout in the Pea Ridge iron mine [5]. The Probable Stope algorithm provides a 2D analysis and enjoys complying with all underground mining methods used for tabular deposits [6]. A Visual C++ program is developed to implement the Probable Stope algorithm, which is called Stope Boundaries Optimizer (SBO). This paper, describes the SBO program in details and the corresponding algorithm briefly.

2. The Probable Stope Algorithm

The Probable Stope Algorithm (*PSA*) is based on the Dynamic Programming technique, which is supported by mathematical proof. The algorithm is applied on a 2D conventional economic model. So, it is best suited for vein type or tabular deposits. In fact, characteristics of the ore body in the third dimension are projected on the model plane, i.e. composited into one block. The *PSA* complies with all underground mining methods used for tabular deposits [6].

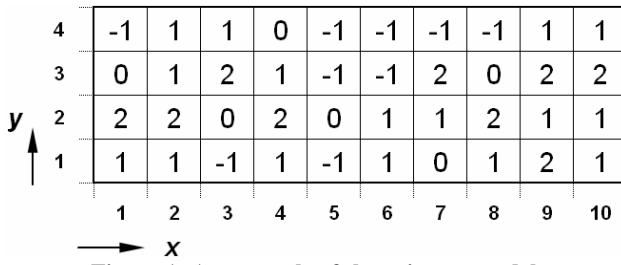
2.1 The Modeling of the Mining Area

The *PSA* applies on a 2D conventional fixed economic block model of the mining area, which is constructed using the geologic, technical and economic data, collected through the conceptual design. This is called, in this paper, the primary model. Then through two stages, the probable stope (*PS*) model and the integrated probable stope (*IPS*) model are derived from the primary model. The algorithm is then implemented on the *IPS* model.

1) The Primary Model

The primary model is a 2D conventional fixed economic block model, which may be thought of as a 2D matrix, where each entry, $m_{y,x}$, represents the net economic value of a block located in the y^{th} row and the x^{th} column. The matrix has X rows representing the total number of blocks in the direction of ore body's strike and Y columns representing the total number of blocks in the ore body's dip direction.

Figure 1 illustrates a simple example of such primary model containing 10 blocks in the strike direction and four blocks in the dip direction.



II) The PS Model

In the *PS* model the stope height constraint is considered. A probable stope in each column includes a set of adjacent blocks in the same column of the primary model which satisfy the minimum height of the stope. In order to make the *PS* model, all probable stopes in each column that satisfy the minimum height constraint, are reviewed and their total economic values are calculated. The number of such probable stopes in each column of the model depends on the stope height constraint (i.e. the restriction in the dip direction).

Consider that the stope height is restricted to a minimum of *h* blocks, the total number of probable stopes may be obtained through Equation (1):

$$NS_{x,h} = \sum_{y=h}^Y (y - h + 1) \quad , \quad 1 \leq x \leq X \quad (1)$$

where $NS_{x,h}$ is the number of probable stopes with a minimum of *h* blocks located at the x^{th} column, *h* is the minimum stope height measured in the number of blocks, *x* is the column number, *X* is total number of columns, *y* is the row number and *Y* is total number of rows.

The location of each probable stope in each column, *x*, is defined as $SL_{(f,c),x}$, where *SL* is the stope location and *f* and *c* which form a pair, represent the location of the floor and ceiling of the stope in column *x*. For example, if the stope contains three adjacent blocks from the first, second and third rows of column *x* (i.e. blocks with net economic values $m_{1,x}$, $m_{2,x}$ and $m_{3,x}$), then its location is addressed by $SL_{(0,3),x}$. Figure 2 shows the location of each of the three probable stopes in column *x* for a model with four rows.

The net economic value of each stope may be obtained through Equation (2):

$$M_{(f,c),x} = \sum_{y=f}^c m_{y+1,x} \quad (2)$$

For $1 \leq x \leq X$, $0 \leq f \leq Y-h$, $h \leq c \leq Y$

Where $M_{(f,c),x}$ is the net value of the stope located at

$SL_{(f,c),x}$ and $m_{y,x}$ is the net value of the block located at row *y* and column *x* of the primary model.

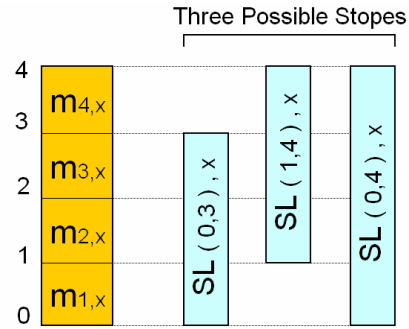
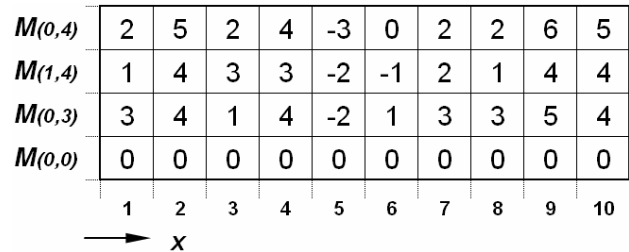


Figure 2: Location of probable stopes in the x^{th} column of primary model [6]

Once the net value of probable stopes are calculated, the probable stope model may be constructed. The total number of columns in this model is the same as that of the primary model and its total number of rows is one more than the number of probable stopes in each column (i.e. $NS_{x,h+1}$). Figure 3 shows the probable stope model of the example shown in Figure 1, assuming a three block height restriction. The lowest row indicates the net value of stopes at the location $SL_{(0,0),x}$, which are in fact virtual stopes with zero (block) height. In fact the *PS* model contains, in each row, the economic value of one probable stope ($M_{(f,c),x}$).



III) The IPS Model

The *IPS* model is constructed by imposing the minimum stope length constraint. The minimum length of the stope depends on factors such as the mining method and the equipment used.

Each block of the *IPS* model contains the maximum economic value of the stope ending at that block, while satisfying the minimum length constraint. If the minimum stope length equals *l* blocks, *l* virtual columns are added to the left of the *PS* model to provide feasible stopes, otherwise none of the blocks located in the first *l* blocks of the model may form a stope with a minimum length of *l* blocks. The net value of all stopes, which are not entirely

located inside the primary model (those with columns $j=1$ to $j=l$) is set to a very large negative value.

Another restriction that must be considered is the maximum variation of the elevation of both the floor and the ceiling of the stope from one column to the next. This constraint is expressed as the number of blocks. Consider that $SL_{(f,c),x}$ represents the stope location at column x , then the stope location at column $(x+1)$ may assume one of the under the following conditions:

$$f + rf \geq 0, c + rc \leq Y, (c + rc) - (f + rf) \geq h$$

$$|rf| = 0, 1, 2, \dots, n_f, \quad n_f < h$$

$$|rc| = 0, 1, 2, \dots, n_c, \quad n_c < h$$

Where rf is an integer number representing the floor level variation constraint, rc is an integer number representing the ceiling level variation constraint, n_f is maximum number of blocks of the floor level variation constraint, n_c is maximum number of blocks of the ceiling level variation constraint.

When dealing with variation in the stope levels, care should be taken not to violate the minimum height constraint. The expression $(c + rc) - (f + rf) \geq h$ has been included in the list of constraints above to guarantee that this condition is satisfied. Imposing the level variation constraint, the maximum net value of stopes with the minimum allowed length may be calculated through the following relations [6]:

$$\text{if } f = 0, c = 0 \rightarrow MSV_{(0,0),j} = 0 \tag{3}$$

$$\text{if } f \neq 0, c \neq 0$$

$$MSV_{(f,c),j} = \begin{cases} -u & \text{Where } 1 \leq j < l \\ M_{(f,c),j-l+1} + \sum_{r=1}^{l-1} \max \{ M_{(f+rf,c+rc),j-l-r+1} \} & \text{Where } l \leq j \leq J \end{cases} \tag{4}$$

Subject to:

$$f + rf \geq 0, \quad c + rc \leq I, \quad (c + rc) - (f + rf) \geq h$$

$$|rf| = 0, 1, 2, \dots, n_f, \quad |rc| = 0, 1, 2, \dots, n_c$$

where $MSV_{(f,c),j}$ is the maximum economic value of the stope with minimum allowed length, ending to $SL_{(f,c),j}$, u is a large positive number, l is the minimum stope length allowed, $M_{(f,c),j}$ is economic value of the stope at $SL_{(f,c),j}$ and r is a counter.

In fact, each cell of the *IPS* model represents the maximum net value of a stope with the minimum length and ending to that cell. Figure 4 illustrates the final model corresponding to the *PS* model of Figure 3.

2.2 The PSA Logic

The *PSA* is proposed to optimize the stope boundaries, which is run on the *IPS* model described above. A Dynamic Programming technique is employed using a recursive formula, similar to that used for open pit limit optimization. However, the formula consists of two criteria. By applying the algorithm to each stope at a specified location, the economic value of all probable mining areas ending to that stope are calculated and the maximum one is specified for the *IPS* model.

Figure 4: The *IPS* model developed using Figure 3

The recursive formula is as follows [6]:

$$P_{(0,0),j} = \begin{cases} 0 & \text{Where } 1 \leq j < l \\ \max \{ P_{(f+rf,c+rc),j-1}, P_{(0,0),j-1} \} & \text{Where } l \leq j \leq J \end{cases} \tag{5}$$

$$P_{(f,c),j} = \begin{cases} -u & \text{Where } 1 \leq j \leq l \\ \max \{ P_{(f+rf,c+rc),j-l}, P_{(0,0),j-l} \} & \text{Where } l < j \leq J \\ + MSV_{(f,c),j} + NSVP_{(f,c),j} & \end{cases} \tag{6}$$

Subject to:

$$f + rf \geq 0, \quad c + rc \leq I, \quad (c + rc) - (f + rf) \geq h$$

$$|rf| = 0, 1, 2, \dots, n_f, \quad |rc| = 0, 1, 2, \dots, n_c$$

where $P_{(f,c),j}$ is the maximum net value of the mining area ending to the stope located at $SL_{(f,c),j}$, $P_{(0,0),j}$ is the maximum net value of the mining area ending to the stope located at $SL_{(0,0),j}$ and $NSVP_{(f,c),j}$ is new stope value parameter.

The value of $NSVP_{(f,c),j}$ is calculated only at two circumstances. The first one is when the algorithm starts and there is no stope formed yet. The second one is when the limits of the stope is passed and a new stope is going to be formed. The value of $NSVP_{(f,c),j}$ is assigned zero, otherwise [6].

Figure 5 illustrates the application of the algorithm on the final model for $n_f=0$, $n_c=1$ and $l=3$, with the corresponding limits. As Figure 5 shows, the optimum limits provide the maximum value of 32 units [6].

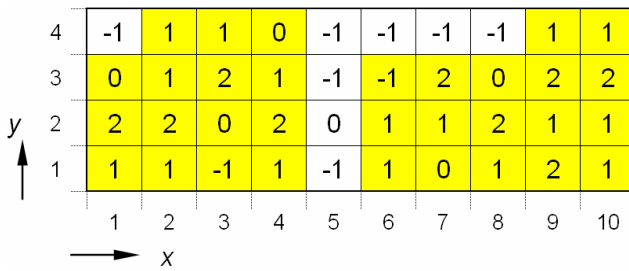


Figure 5: An application of the PSA [6]

3. The SBO Program

A computer program is developed to implement the *PS* algorithm described above, using Visual C++ computer programming language. The program is called Stope Boundaries Optimizer (SBO); the program is implemented on a 2D conventional economic block model of the ore body to locate the optimum mining area, which provides the maximum profit. The algorithm, which is the base for developing the SBO program, is capable of defining the constraints imposed to all conventional underground mining methods used for mining tabular deposits. These constraints include the minimum length and height of the stope.

The structure of the SBO program is designed such that the optimization process is divided into a number of steps. Firstly, the *PS* model is constructed. Then, the *PS* model is modified to build the *IPS* model and record all relevant information. Next, the Probable Stope algorithm is run over the *IPS* model. The implementation begins from the left of the *IPS* model and continues to the right. During the run of the program, the information resulted from processing the model, which shows the economic values of the processed areas are recorded for further use. The SBO may be run on usual personal computers.

3.1. The Input Data

The input data related to the primary model are provided with a text file, consisted of three columns. These columns represent the X-coordinate, Y-coordinate and the net economic value of each block (BEV) of the primary model. Table 1 illustrates an example of the input file to the SBO program. These pieces of information are provided based on the economic model shown in Figure 2 and are saved as a text file called "INPUT01.DAT". (For saving the space, some parts of this file have been omitted). The SBO input file may be saved as any name and extension.

Table 1: The SBO input file corresponding the primary model, shown in Figure 1

File ame: INPUT01.DAT		
X	Y	Values
10	15	1
20	15	1
30	15	-1
40	15	1
:	:	:
40	45	1
50	45	-1
60	45	-1
70	45	2
:	:	:
70	60	-1
80	60	-1
90	60	1
100	60	1

After saving the data related to the economic model in a text file, the program may be run to complete the input information. The SBO program receives three types of information through a dialogue box menu formed by visual C++ facilities. The first group includes the general information such as the project title, the name of the input file, in which block addresses and economic values are stored and the name of the output file. The second type of input information is related to the structure of the primary model and the system of addressing in the input file, including the distance between two adjacent blocks (centre to centre), in both the strike and dip directions. The third type of information refers to the stope specifications based on the selected mining method, including the minimum length and height constraints of the stope and the maximum variation of the elevation of both the floor and ceiling of the stope from any column to the next one.

Figure 6 shows the sub-menu of the SBO menu-bar for entering the input data and type of output data. In the first section, the names of both the SBO input and output files are presented for the example, shown in Figure 1.

Block dimensions in the strike and dip directions have been assigned 10 and 15 units, respectively. The minimum stope length and height are both assumed to be three blocks. The maximum variations in the level of the floor and ceiling are assigned zero and one block, respectively.

After entering the first group of information, the preliminary process of information is begun. The process includes reading the input file and

controlling the recorded information. If there is any mistake in the input data, a convenient error message is prompted to the user for any data correction. The most important mistakes here are finding two or more records with various values but identical addresses.

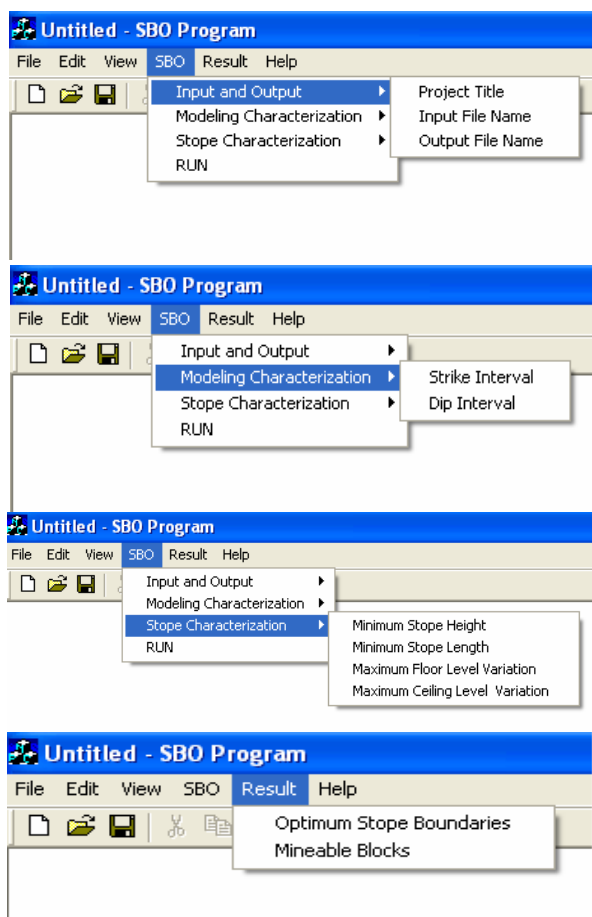


Figure 6: The SBO menu for entering input data and selecting output format.

After it is made sure that the input data are error-free (data verification), the program receives the second type of information. Then, the SBO proceeds to process the second type of data. The input data are checked to be error-free and a convenient message is prompted to the user to either correct the input data or proceed to the next step. The most important cause of halting the process is lack of compatibility of the size of the blocks with the size of the model in both the *X* and *Y* orthogonal directions, which results in non-integer number of blocks. If so, an adjustment should be made to correct the input data. Provided that input data are verified and acceptable, the program computes the ranges of values that the minimum stope length and height and the maximum variation of the stope level

(for both the floor and the ceiling) may assume. These ranges are then given to the user to select a choice among them. The user may then enter the third group of information.

Once all groups of information are entered through the main menu, the SBO program creates the output file and prompts the name and path of the output file to the user.

3.2. The Output Results

After all required data are entered and verified, the SBO creates the primary, *PS* and *IPS* models as some matrices to find the economic values of various mining areas. These matrices are created to; only, provide the means to control the correctness of the input data. All results are stored in a file, whose name and extension are already defined through the main menu, shown in Figure 6. As shows in the figure, two formats are available to from the results and user may select one of them. Example of these pieces of information recorded in the output file (OUTPUT01.DAT) excluding the primary, *PS* and *IPS* matrices, generated by SBO, are shown in Table 2 and 3.

The most important expected results of using SBO are the determination of the optimum boundaries of the mining area and its net value. In order to find the corresponding limits, the SBO program generates the matrix, shown in Table 2. This matrix corresponds the primary model as well and its entries have the value of "1" or "0". All blocks that should be mined to satisfy the constraints and form the optimum stope boundaries are assigned the value "1". Other blocks that should be left unmined are assigned the value "0". Therefore, the ultimate stope boundaries are defined on the primary model. The output file of the SBO program also contains a list of blocks, which fall within the optimized boundaries. The list includes information such as the *X* and *Y* coordinates and the net economic values of the blocks. Table 3 shows an example of this list, presented at the end of the SBO output file. As shown, the total value of the mining boundaries equals 32 units.

Table 2: The optimized stope boundaries

		The Optimum Stope Boundaries									
		1	2	3	4	5	6	7	8	9	10
4		0	1	1	1	0	0	0	0	1	1
3		1	1	1	1	0	1	1	1	1	1
2		1	1	1	1	0	1	1	1	1	1
1		1	1	1	1	0	1	1	1	1	1

Table 3: List of blocks of the optimum slope boundaries

The Mineable Blocks		
X	Y	Value
100.00	60.00	1.00
100.00	45.00	2.00
100.00	30.00	1.00
100.00	15.00	1.00
90.00	60.00	1.00
90.00	45.00	2.00
90.00	30.00	1.00
90.00	15.00	2.00
80.00	45.00	0.00
80.00	30.00	2.00
80.00	15.00	1.00
70.00	45.00	2.00
70.00	30.00	1.00
70.00	15.00	0.00
60.00	45.00	-1.00
60.00	30.00	1.00
60.00	15.00	1.00
40.00	60.00	0.00
40.00	45.00	1.00
40.00	30.00	2.00
40.00	15.00	1.00
30.00	60.00	1.00
30.00	45.00	2.00
30.00	30.00	0.00
30.00	15.00	-1.00
20.00	60.00	1.00
20.00	45.00	1.00
20.00	30.00	2.00
20.00	15.00	1.00
10.00	45.00	0.00
10.00	30.00	2.00
10.00	15.00	1.00
Total Value=		32.00

4. A Numerical Example

In order to validate the SBO program, a model section previously optimized using other techniques, is studied. Figure 7 shows a 2D economic block model employed by Riddle (1977) to optimize the slope limits, using a Dynamic Programming algorithm, provided that the deposit is mined using block caving method.

Mining constraints were assumed the same as what used by Riddle to compare the results. The minimum length and height constraints were assigned two blocks and the maximum variation in the floor and ceiling level were assigned zero and one block, respectively.

The provided information about the primary model is recorded in a file named INPUT02.DAT and the input data are shown in Table 4.

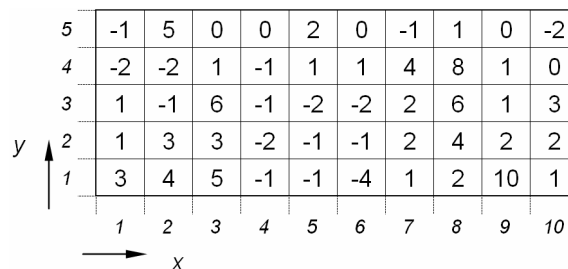


Figure 7: A worked example of the primary model

Some parts of the results for the worked example, recorded in the SBO output file, are shown in Table 5 and 6.

Table 6 shows the optimized slope boundaries as a matrix corresponding the primary model by “1” and “0” entries. All entries with value of “1” refer to blocks that should be mined and other entries that have the value of “0” indicate to blocks that should be left in modeled mining area.

Table 6 shows a list of blocks that fall within the optimized boundaries. As shown in the end of the file, the maximum value of the stop boundaries equals 77 units.

In order to compare the results obtained using SBO with those obtained using Riddle’s Dynamic Programming technique, the optimized layout of the slope imposed on the primary model using both alternatives are presented in Figure 8. A comparison between the results shows that using the SBO program provides a slope with a higher net value as well as ore content.

Table 4: The SBO input data for the worked example

Project Title	A Numerical Example
Input File Name	INPUT02.DAT
Output File Name	OUTPUT02.DAT
Strike Interval	1.00
Dip Interval	1.00
Minimum Slope Height	2
Minimum Slope Length	2
Maximum Floor Level Variation Constraint	0
Maximum Ceiling Level Variation Constraint	1

Table 5: Slope boundaries of implementing SBO on the worked example

The Optimum Slope Boundaries

	1	2	3	4	5	6	7	8	9	10
5	0	1	1	0	0	0	0	1	1	0
4	1	1	1	0	0	0	1	1	1	1
3	1	1	1	0	0	0	1	1	1	1
2	1	1	1	0	0	0	1	1	1	1
1	1	1	1	0	0	0	1	1	1	1

Table 6: List of blocks of the optimized stope boundaries for the worked example

The Mineable Blocks

X	Y	Value
10.00	4.00	0.00
10.00	3.00	3.00
10.00	2.00	2.00
10.00	1.00	1.00
9.00	5.00	0.00
9.00	4.00	1.00
9.00	3.00	1.00
9.00	2.00	2.00
9.00	1.00	10.00
8.00	5.00	1.00
8.00	4.00	8.00
8.00	3.00	6.00
8.00	2.00	4.00
8.00	1.00	2.00
7.00	4.00	4.00
7.00	3.00	2.00
7.00	2.00	2.00
7.00	1.00	1.00
3.00	5.00	0.00
3.00	4.00	1.00
3.00	3.00	6.00
3.00	2.00	3.00
3.00	1.00	5.00
2.00	5.00	5.00
2.00	4.00	-2.00
2.00	3.00	-1.00
2.00	2.00	3.00
2.00	1.00	4.00
1.00	4.00	-2.00
1.00	3.00	1.00
1.00	2.00	1.00
1.00	1.00	3.00

Total Value=77.00

5	-1	5	0	0	2	0	-1	1	0	-2
4	-2	-2	1	-1	1	1	4	8	1	0
3	1	-1	6	-1	-2	-2	2	6	1	3
2	1	3	3	-2	-1	-1	2	4	2	2
1	3	4	5	-1	-1	-4	1	2	10	1
	1	2	3	4	5	6	7	8	9	10

Max. Stope Value = 77

a) Optimization layout using SBO

5	-1	5	0	0	2	0	-1	1	0	-2
4	-2	-2	1	-1	1	1	4	8	1	0
3	1	-1	6	-1	-2	-2	2	6	1	3
2	1	3	3	-2	-1	-1	2	4	2	2
1	3	4	5	-1	-1	-4	1	2	10	1
	1	2	3	4	5	6	7	8	9	10

Max. Stope Value = 76

b) Optimization layout using Riddle's program [3]

Figure 8: SBO vs. Riddle's program

5. Conclusions

The SBO program has been written by visual C++ code, based on the probable stope algorithm. The SBO program creates a probable stope model and integrated probable model, respectively, considering the technical constraints of the underground mining method, including the minimum length and height of the stope; determines economically optimum stope boundaries using the probable stope algorithm; and records the outputs in a data file, including the maximum values of all probable stope boundaries, the address of all blocks, which fall within the optimum limits and their total value. The SBO complies with all underground mining methods used for tabular deposits. The SBO is user friendly and its application to optimal stope design is very simple and also can be operated with very little training.

6. References

- [1] C. Alford, "Optimization in Underground Mine Design", in *1995 Proceedings of the 25th International Symposium on the Application of Computers and Operations Research in the Mineral Industry*, The Australasian Institute of Mining and Metallurgy, Melbourne, pp. 213-218.
- [2] M. Ataee-pour, "A Heuristic Algorithm to Optimize Stope Boundaries", Ph.D. dissertation, Dept. Mining Eng., University of Wollongong, 2000.
- [3] J. Riddle, "A Dynamic Programming Solution of a Block-Caving Mine Layout", in *1977 Proceedings of the 14th International Symposium on the Application of Computers and Operations Research in the Mineral Industry*, Society for Mining, Metallurgy and Exploration, Colorado, pp. 767-780.
- [4] J. Ovanic, and D. Young, "Economic Optimization of Stope Geometry Using Separable Programming with Special Branch and Bound Techniques", in *1995 Third Canadian Conference on Computer Applications in the Mineral Industry*, H. Mitri (ed.), Balkema, Rotterdam, pp. 129-135.
- [5] J. Ovanic and D. Young, "Economic Optimization of Open Stope Geometry", in *1999 Proceedings of the 28th International Symposium on Computer Applications in the Minerals Industries*, K. Dagdelen (ed.), Colorado School of Mines, pp. 855-862.
- [6] S. E. Jalali and M. Ataee-pour, "A 2D Dynamic Programming Algorithm to Optimize Stope Boundaries", in *2004 Proceedings of the 13th symposium on Mine Planning and Equipment Selection*, (eds. M. Hardygora et al.), Rotterdam, Balkema, pp. 45-52.